

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

# **ACORN Design Concepts**

December 15, 2021

## **Project ACORN**

#### Vision statement:

– The Accelerator Controls Operations Research Network (ACORN) project will modernize the accelerator control system and replace end-of-life accelerator power supplies to enable future operations of the Fermilab Accelerator Complex with megawatt particle beams. The control system will be a unified system that satisfies user requirements, is integrated with and supports operating experiments, is maintainable and adaptable to future needs, and empowers users to achieve their research goals.

### **Design Alternatives**

As a DOE O413.3B project, ACORN is required to evaluate alternatives for the design of the accelerator control system:

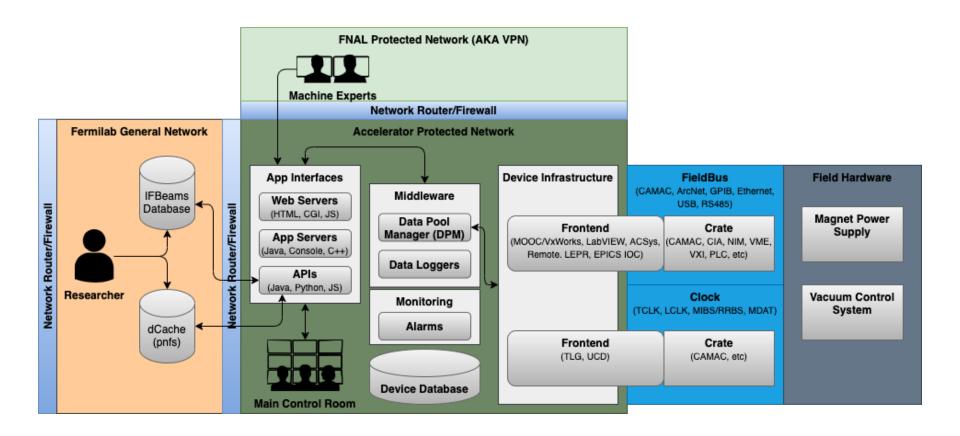
Four alternatives have been specified by DOE:

- Alternative I: Status Quo
- Alternative II: One-For-One Replacement
- Alternative III: Centralized Architecture
- Alternative IV: Decentralized Architecture

#### Alternative I – Status Quo

- Null alternative: do nothing
  - This is the alternative if ACORN does not modernize the accelerator control system
  - Used as a baseline for comparison
- Operational risks
  - Aging technology and obsolescence
  - Lack of expertise due to people leaving the lab
  - Hidden operational costs and risks in the long run

## Alternative 1 high-level block diagram





### **Alternative 1 interface diagram**

#### Alternative 1 - Existing Control System The lines indicate an interface or relationship between systems. Services Beam Sync This services box assumes that each of blue applications that connect to the CLK MDAT outside of the box can be used from outside the box, given a protocol definition Many services are exposed through ACLD, ACNETD, or DPM. SYNC UCD Dedicated links UDP/MC/UCD Synoptic (Outside Access) ACNETD ACSvs FE acl.pl UDP/ACNET/SCALING Scaling UDP/MC/UCD Python Apps UDP/MC/UCTUDP/ACNET/ ACSys Proxy UDP/ACNET/DPM UDP/MC/UCD TCPIWS1 Data Logger UDP/ACNET UDP/ACNET/ ACLD JavaScript Apps LIDP/ACNET Bunny UDP/ACNET/ Inner Apps EPICS IOC USBIRS485IVME] UDP/ACNET TCP/MO TCP/RMI TCP/ACNET UDP/ACNET TCP/RMI Consolidat Fortran / C / C++ / Erlang Apps UDP/MC/ RACKMN -UDP/ACNET ReadingJob LOOKUP UDP/ACNET/ OAC Logger UDP/ACNET/ FShare TCP/IRMIIDPM1 TCP/MQ File Share DownloadD (dnldd) RabbitMQ/Broker TCP/MQ UDP/ACNET UDP/MC/ACNET/STATES SybSet TCP/MQ Java Apps CLKLOG DBServlet Web App LabVIEW Frontend (TCPIUSB) NODESD CPLD (Talk to Jim S.) UDP/ACNET/CPLD TCP/NFS JDBC(TCP/SQL) LBOE NFS (Talk to Jim S., Michele M-W) CLIBLIDBC/TCP/SOLV IRM downld (IRM Dabble SRM The databases here are actually tables in the central Sybase database The exception is the data logger data which is a MySQL database. Add DABBEL for the device database Line Labels CPLD Clock The line labels indicate the layers of communication protocols defined by the interface. Left are lower (outer) level protocols while right are higher (inner) level protocols (OSI model). Square brackets with a bar (I) separator consolidate multiple protocols at the Square brackets with a bar (i) separator consolidate multiple protocols at the same level. Paraenthesis indicate that the protocol level communication is handle by a common library. Wildcard (') indicates that any protocol can be used. Data logger

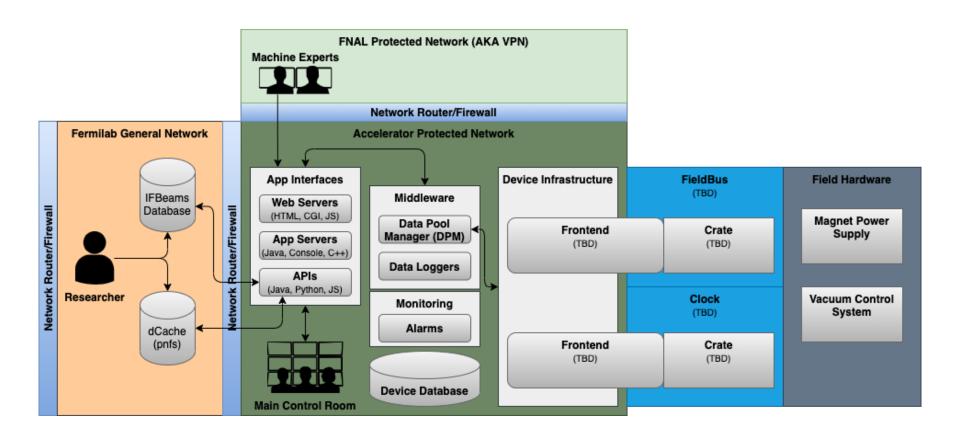


## Alternative II – One-For-One Replacement

- Operations risks are addressed in the context of the current control system architecture
  - Hardware is replaced with newer versions, as equivalent as possible
  - Software gets refactored to use newer technologies / frameworks / programming languages, keeping their functionality and interfaces
  - Interfaces are kept wherever is feasible
    - Protocols between components
    - Links / communication channels
    - User applications
    - Central infrastructure
  - Still need same expertise as Alternative #1
- Hidden operational costs and risks in the long run



## Alternative 2 high-level block diagram



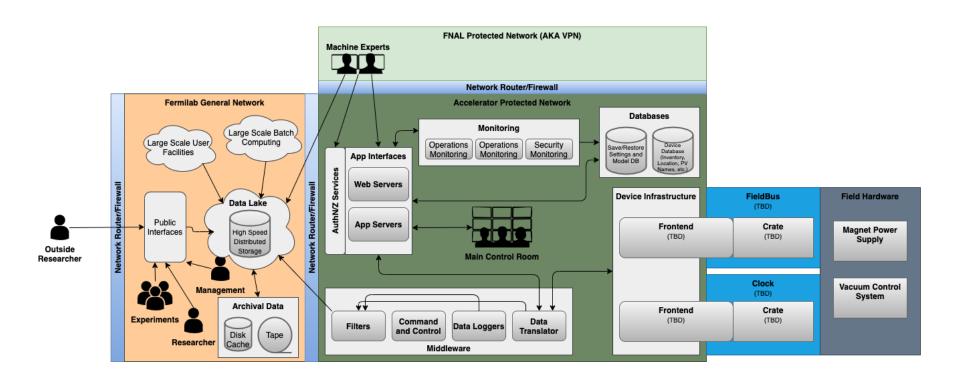


#### **Alternative III – Centralized Architecture**

- Rearchitect the control system based on a centralized architecture
  - Centralized command and control logic
  - Centralized data logger service / data translation
  - Centralized monitoring
  - Uniform interface for control and data acquisition
    - Everything "funnels" through centralized infrastructure (simple networking)
    - Complexity is handled by the centralized infrastructure (can add new services with less impact)
    - Consistent feature set across systems (vertical scaling)
- Remove aging technology
- Minimize the amount of specialized expertise needed to maintain and operate the accelerator and control system
- Eliminate hidden operational costs and risks in the long-term



## Alternative 3 high-level block diagram



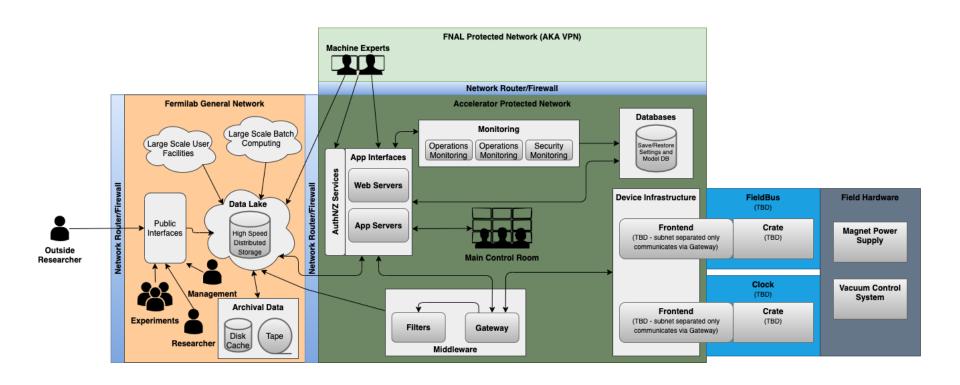


#### **Alternative IV – Decentralized Architecture**

- Rearchitect the control system based on a decentralized architecture
  - Command and control logic is distributed among front-ends
    - Data logger and data translation
    - Complex networking
    - Adding front-ends is a simpler process (horizontal scaling)
  - Client applications contain inherent knowledge of the physical implementation of the control system
    - Know what data is available and how to query for the data
- Remove aging technology
- Minimize the amount of specialized expertise needed to maintain and operate the accelerator and control system
- Eliminate hidden operational costs and risks in the long-term



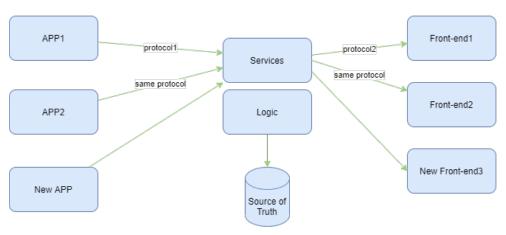
## Alternative 4 high-level block diagram





#### **Centralized vs Decentralized Architecture**

#### **Centralized Architecture**



#### **Decentralized Architecture**

